NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

OPEN-SOURCE DATA COLLECTION TECHNIQUES FOR WEAPONS TRANSFER INFORMATION

by

Frederick C. Krenson Jr.

March 2012

Thesis Advisor: Neil C. Rowe
Second Reader: Joel D. Young

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2012	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Open Source Data Collection Techniques for Weapons Transfer Information			**5. FUNDING NUMBERS**
6. AUTHOR(S) Frederick C. Krenson Jr.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ______NA__________.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Monitoring weapons proliferation requires detecting and tracking weapons transfers. Many public sources of weapons-transfer data come from incomplete and manually collected and maintained government and commercial records. We propose a technique to gather weapons-transfer information by mining publicly available Web pages for features, which we categorize as arms, actions, actors, or money. We design a retrieval system and parser, and develop techniques for extracting currency values from text, measuring precision without available training data, and measuring recall with a parallel but different corpus. Results show that, of the sentences matching four feature categories, 70% of relevant features were found, and sentences that only matched three categories introduced more false positives. We conclude that such a technique can improve the speed at which transfer information is compiled.

14. SUBJECT TERMS Data mining, crawler, natural language processing, weapons			**15. NUMBER OF PAGES** 73
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

OPEN-SOURCE DATA COLLECTION TECHNIQUES FOR WEAPONS TRANSFER INFORMATION

Frederick C. Krenson Jr.
Civilian, SPAWAR Systems Center Atlantic
B.S., Clemson University, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2012

Author: Frederick C. Krenson Jr.

Approved by: Neil C. Rowe
Thesis Advisor

Joel D. Young
Second Reader

Peter Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Monitoring weapons proliferation requires detecting and tracking weapons transfers. Many public sources of weapons-transfer data come from incomplete and manually collected and maintained government and commercial records. We propose a technique to gather weapons-transfer information by mining publicly available Web pages for features, which we categorize as arms, actions, actors, or money. We design a retrieval system and parser, and develop techniques for extracting currency values from text, measuring precision without available training data, and measuring recall with a parallel but different corpus. Results show that, of the sentences matching four feature categories, 70% of relevant features were found, and sentences that only matched three categories introduced more false positives. We conclude that such a technique can improve the speed at which transfer information is compiled.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AECA	Arms Export Control Act
API	Application Programming Interface
ASP	Active Server Pages
DNS	Domain Name System
DoD	Department of Defense
DOM	Document Object Model
DPRK	Democratic People's Republic of Korea
EU	European Union
FTP	File Transfer Protocol
GBP	Pound sterling, British currency
GHz	Gigahertz
GNU	GNUs Not Unix
GUI	Graphical User Interface
HMM	Hidden Markov Model
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over SSL
IE	Information Extraction
I/O	Input/Output
IP	Internet Protocol
IR	Infrared
ISO	International Organization for Standardization
ITAR	International Traffic in Arms Regulations
NER	Named Entity Recognition
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
OWL	Web Ontology Language
PAIGC	African Party for the Independence of Guinea and Cape Verde
PHP	PHP Hypertext Preprocessor
RAM	Random Access Memory

RDF	Resource Description Framework
SAM	Surface-to-Air Missile
SIPRI	Stockholm International Peace Research Institute
SQL	Structured Query Language
SSNE	Semi-Structured Named Entity
SVM	Support Vector Machine
TCP	Transmission Control Protocol
UAE	United Arab Emirates
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USSR	Union of Soviet Socialist Republics
UTF	Unicode Transformation Format
USML	United States Munitions List
W3C	World Wide Web Consortium
XML	Extensible Markup Language

ACKNOWLEDGMENTS

I would like to thank Dr. Neil Rowe and Dr. Joel Young for their help and guidance while completing this work. I would also like to thank SPAWAR for funding my degree.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Weapons proliferation is a global concern. It can change the defensive posture of an actor, and can be destabilizing between actors with regional, political, or military ties. Actors in this context include nations, organizations, and individuals seeking to acquire or distribute weapons. We are most interested in nations in this study as national governments tend to have the capital necessary to make larger weapons acquisitions. Nations also have a legitimate need to maintain armed forces requiring the production or acquisition of weapons. To better understand the threats posed by arms transfers and proliferation, policy makers in the United States are provided with intelligence to help develop domestic and foreign policy. The United States Munitions List (USML) contains 21 categories and is a component of the International Traffic in Arms Regulations (ITAR) [1], which set policy for imports and exports of controlled arms. The authority to monitor and regulate these transfers is granted by the Arms Export Control Act (AECA) [2] and is implemented by the ITAR. With these regulations, all arms with a military application must be properly documented and approved before import or export. Actors with hostile intentions may be precluded from dealing with the United States; however, the AECA does not apply to other countries and therefore transfer details must be collected using alternative methods. Intelligence regarding these weapons of interest will give insight to their defensive capabilities as well as international relationships with other actors.

Many reports concerning controlled arms are compiled by the Congressional Research Service for members of Congress and include information about transfers that do not involve the United States. One such report covering weapons sales to developing nations details the types of weapons, total transfer value by country, countries they may have traded with, and an analysis of the past eight years of activity [3]. Similar analysis is available from several sources although not all of the methods used to aggregate relevant are known, and in some cases [4], [5], [6] the data is secured from official government reports, or user-submitted forms. If the acquisition of weapons transfer

activity could be automated it would reduce the need for previously manual efforts, reveal gaps in current knowledge, and shed light on historical and emerging trends.

There are several challenges associated with locating, accessing, storing, and interpreting relevant content on the Web. The number of accessible Web pages varies greatly depending upon the test method and source of information but, as of 2005, this number was estimated to be in the billions [7]. Search companies like Google have already indexed large portions of the Web and we can capitalize on these results to narrow our search domain. Google no longer reveals its search index size, claiming that there are too many factors that affect the accuracy of the number; however, Google does claim that its indexes are over three times larger than those of its competitors [8]. Still, the percentage of Web pages that contain information related to weapons sales is unknown. Search engines do not guarantee that results for a specific query will be relevant or current. Because of this, aggregating information on a specific topic requires more work than analyzing keyword indexes and page ranks. These may help in determining a statistical likelihood that the content of a Web page is relevant to a given query; however, extracting specific details from these pages is currently an application-specific problem. The variability of Web content also complicates this task. Examples of situations that may impede access to information and make pattern recognition difficult include misspellings and other typographical errors, colloquialisms, unsupported encodings, and scripting technologies. There may also be a widely-accepted variation on the formatting of common items. For instance, monetary values can be formatted in many ways. To assist with the problems presented by this, a grammar was developed to help identify valid forms. Another common Web design is to utilize JavaScript (or other scripting technologies) to dynamically control a Web page. Simply downloading the page may not yield the content that was intended to be displayed; the script must be executed by the client's browser first. These are only a few of the complications that should be addressed when accessing Web data.

In this study, we would like to hypothesize that an automated process for obtaining and reporting on transfers will be more efficient than a manual process. To accomplish this, we develop a prototype tool to automatically locate, acquire, and parse

weapon transfer information. We then attempt to answer two questions: (1) What is the performance, precision, and recall of this technique? (2) What interesting correlations can be made using the data that has been retrieved and parsed? We do not attempt to determine the accuracy of the data that is acquired, as that is a research topic in itself, although we acknowledge that not all such information is correct. We confine ourselves to showing how this information could be obtained automatically using search engines, and we attempted to identify the actors involved, the weapons being transferred, the monetary value involved, and the type of transfer using natural-language processing techniques on a set of Web results. Though we attempt to identify a monetary value, the acts we find may not involve an exchange of money. After identifying each of these features, we show statistics on how these features may be combined to identify relevant information and correlations.

This study is organized into five chapters. Chapter II covers background and related work on search engines, natural-language processing, and the Semantic Web. Chapter III covers the methodology that includes the program architecture, storage model, crawler, feature identification and how features were combined to recognize relevant content. Chapter IV is a discussion of the results that were recorded using this application, and Chapter V is the conclusion.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND AND RELATED WORK

Data mining of information from the Internet has been practiced for twenty years now. This work draws upon information and techniques currently used in industry. These techniques include search-engine design, and natural-language processing techniques.

A. WEB TECHNOLOGIES

The World Wide Web is defined by the W3 Consortium as "an information space in which the items of interest, referred to as resources, are identified by global identifiers called Uniform Resource Identifiers (URI)" [9]. These URIs are also known as Uniform Resource Locators (URL). URIs are typically a domain and resource name paired together to identify the host, or Web server, and specific content of interest on that host. Users interact with these hosts through common protocols such as the Hypertext Transfer Protocol (HTTP) which is an application-layer protocol. HTTP defines many commands for transferring resources between two entities, some of the most common methods being GET and POST. GET is a request to a Web server to transfer a specific resource back to the requestor. POST serves to transfer a resource from the requestor to a Web server. Each command will result in a status code being returned indicating success or failure of the operation. For example, HTTP 200 notifies the requestor that the command succeeded while HTTP 404 indicates to the user that the requested resource identified in a GET command was not found [10].

The resources commonly transferred are Hypertext Markup Language (HTML) files. HTML describes the formatting of the Web page when it is viewed in a browser. The formatting is accomplished with the use of tags. The tags can control several parts of a page including the size, and color of text, and the location and size of images among many other things. Current Web technologies go beyond formatting content on Web pages, and also employ scripting languages to provide more functionality. Server-side scripting languages such as PHP, ASP, process content either before or after it has been

delivered to the requestor, where client-side scripting languages such as Javascript are sent to the requestor for execution. Both scripting types can allow for more interactive and customized Web browsing.

B. SEARCH-ENGINE DESIGN

Search engines identify World Wide Web content we are interested in and it is helpful to understand how they operate. There are many ways to construct a search engine. Arasu et al. [11] propose a general search engine framework including crawlers and crawl-control engines, an indexer module, a collection analysis module, page repositories and indexes, a query engine, and a ranking system. The crawlers and crawl control engines work together to download content (Web pages) and control how the crawler behaves, including what order to investigate pages among other parameters. The indexer module maintains a keyword index, or lookup table to pages. The collection-analysis module provides lookup capabilities and keyword-to-URL mappings. The page repository is the local set of content that was downloaded by the crawler, and the indexes are the lookup tables generated by the indexer module for each page. User requests are executed by the query engine, which uses the prebuilt indexes to identify related content. The results returned are usually ranked by some form of a ranking engine, which estimates which pages are most relevant to what was submitted in the query.

1. Crawlers

Crawlers are utilities that are used to retrieve Web content. Popular standalone crawlers include GNU Wget [12] developed by members of the Free Software Foundation, and cURL [13]. Such programs can be executed through the command line, a script, or a GUI. Libraries such as libcurl [13] and httplib2 [14] are also available which simplify the downloading of Web content. These libraries allow developers more fine-grained control of how content is retrieved and how it is handled upon download.

Crawlers have been developed to handle a wide variety of protocols (e.g., HTTP, FTP), on many operating systems. The cURL program (and library) supports 21 different protocols, and 31 different operating systems alone. Httplib2 is a module and works on any OS that can run the necessary version of Python and supports only HTTP and

HTTPS. A good crawler will provide detailed connection information, and options controlling the behavior during various protocol states. An example of this would be how a crawler could be programmed to handle a redirect notice (e.g., HTTP 307 temporary redirect.)

2. Deep Web Crawling

Some Web pages return content only when queried in a specific manner. Deep Web crawling involves identifying these protocols and developing techniques to implement them via some form element and/or script that retrieves information from a database for display. It has been estimated that 400–550 times more public data is available via the deep Web than that of the "commonly defined World Wide Web" [15]. But obtaining data from such pages is difficult because each Web site may have different form variables, database structures, authentication schemes, or other aspects which require a different retrieval plan. It may also take numerous queries to such a page to yield all of the content that can be delivered. This particular study does not focus on the deep Web but addressing it is a possible extension of our work.

3. Content Freshness

Being able to return relevant (and usually current) data is a big goal in designing a search engine. To ensure that a crawler maintains an up-to-date repository, the content of changing pages must be refreshed. Measuring content freshness entails analyzing last-visited time, and comparing it to the frequency for which those pages should be visited. Two options for the refreshing presented by [11] are a uniform policy, and a proportional policy based on how frequently a page changes. The proportional policy can be difficult to determine if changes are not recorded and published. The uniform policy refreshes all content at predefined intervals. A technique to estimate a better refresh frequency based on the number of detected changes to content was presented by [16]. When this technique was simulated with two different estimation functions it resulted in a 193% and 228% increase in the detection of changes respectively over the uniform refresh policy.

4. Crawling Techniques

Crawlers are given an initial list of URLs to visit. From this list, they initiate connections to Web sites, download pages for analysis, and start populating indexes and discovering other pages to visit. A crawler will usually be programmed to prioritize the next pages to visit using some method. This method may be based on several factors, such as content freshness or relevance. Google uses a technique called PageRank to estimate the number of times a page has been linked to from another location on the web [17]. This assists not only with identifying which pages may warrant a more frequent refresh rate, but also in returning more relevant results to a given search query. After a crawler prioritizes the URLs to visit, the next step is to connect to Web sites to download content at a rate which does not consume significant bandwidth or resources on the remote server. One convention to control how crawlers access Web sites is through the Robot Exclusion Standard [18]. Server operators can place a robots.txt file at the root of their site that will affect the operation of crawlers which are sophisticated enough to interpret it.

C. NATURAL-LANGUAGE PROCESSING

Natural-language processing involves automating the analysis of human languages (e.g., English and Spanish). This study analyzes the textual content of Web pages and thus will focus on natural-language processing. We assume that multimedia (e.g., audio and video) is not particularly helpful to locate information of interest concerning arms sales.

1. Information Extraction (IE)

There are several methods for extracting information and meaning from text. Common kinds include Named Entity Recognition (NER), relation detection and classification, event detection and classification, and temporal analysis [19]. Named entity recognition identifies specific items such as people, places, and things. Relation detection and classification involves finding phrases such as “part of” or “located near” linking two or more objects. Event detection and classification correlates references to

events and to certain entities in the text. Temporal analysis detects and converts time references within text. It can be used to estimate a date the article was made, or find dates within an article.

The ability to discover new word relationships from corpora is a fundamental aspect of natural-language processing. Hearts [20] discusses a technique for mapping hyponyms within a corpus using lexico-syntactic patterns e.g., “such as,” “including,” “or other,” “and other,” and “especially.” The techniques for detecting specific entities in text have improved over many years, and many have resulted in a higher precision and recall due to their ability to handle larger variations of language and formatting within corpora. One such method proposed by [21] proposes a three-phase bootstrapping system for analyzing text and using machine learning algorithms to build a final list of patterns for detecting Semi-Structured Named Entities (SSNE), which include items such as phone numbers, dates, and times. The authors argue that approaches such as regular expressions, and machine learning alone are inefficient in processing their tasks due to the former being too large and unwieldy, and the latter needing a large set of data to assist with supervised learning. In this thesis we present a method for detecting valid forms of currency which we define using a grammar and implement the checks using a set of regular expressions. Though [21] argues that regular expressions are an inefficient method to perform this task for general data mining, they can be efficient for narrower data-mining tasks such as ours and the grammar we have built can be easily adapted to new currency formats.

In other works, such as [22], hidden Markov models (HMM) are used to identify named entities and properly classify them within biomedical texts. They compute word similarity from large, unlabeled corpora in the form of word proximity relationships and use these statistics to assist with analysis when there is not enough data to determine the correct context. Another approach to performing NER shown by [23] is to use a maximum entropy algorithm. The idea is to determine probabilities with as few assumptions as possible, only using the properties derived from an available training set as constraints. An implementation for NER with a Support Vector Machine (SVM) Lattice was proposed by [24]. The lattice uses an HMM to determine the probability of

word tags within a sentence, and a SVM as a binary classifier to help train the system to correctly identify language-neutral entities.

2. Morphology

Another important aspect of information extraction is morphology, the "study of the way words are built up from smaller meaning-bearing units, morphemes" [19]. Examples are dealing with singular or plural nouns, past, present, or future tenses of verbs. Some words like "test," could also be used in other forms such as "testing," "tested," and "tester." Morphology attempts to identify stems and affixes; the former being the root of the word and the latter consisting of the prefixes, infixes, suffixes, and circumfixes that can be added to the root word [19]. Simple pattern matching techniques like regular expressions for the variations of target words can be a quick solution if one only wants to detect their occurrence.

3. Tools

Several resources can assist with natural-language processing. One resource is an online database of word senses and relations, called WordNet. This lexical database provides a link between words, their definitions, hyponyms, hypernyms, holonyms, synonyms, and related forms among other things [25]. Programs that can interface with WordNet are available for multiple operating systems, and libraries have been written to facilitate interaction with this database in multiple programming languages as well.

One method to perform lexical analysis is by using the use of the Natural Language ToolKit (NLTK) [26]. This library not only provides WordNet capabilities but also several tools for interpreting text, such as tokenizers and parts-of-speech taggers. The NLTK also provides several corpora and associated parsing tools to give users access to a large set of information [27]. The corpora have different license restrictions that may affect how they may be used. Some may have been released into the public domain, are for non-commercial use only, reference an existing license, specify restrictions for individual files within the corpus, or have no license restrictions documented. Depending on the corpora used, these restrictions can affect their use in publications and derivative works. Though we do not rely on the corpora provided by NLTK, we do utilize the

WordNet lexical database to discover hyponyms for specific words. NLTK also provides several tokenizers to split content on character, word, or sentence boundaries; however, the task of tokenizing sentences requires more consideration. To assist with this task, they include the Punkt tokenizer that can be trained to parse sentences more effectively.

D. SEMANTIC WEB

There are many ways to locate relevant results based upon keyword search and indexing. A search engine may not be able to find matches with an overly specific search query, or may find false matches that were not what the user wanted. The goal of the Semantic Web is "to explicate the meaning of Web content by adding semantic annotations" with the goal of increasing match accuracy [28]. The semantic annotations could be done in any organized format but standards help. One attempt at a standard is the Web Ontology Language [29]. The World Wide Web Consortium (W3C) released version 2 of this language in 2009, the primary exchange syntax of which is RDF/XML. While RDF/XML is the primary format, other ontology exchange formats are also supported as part of this specification. Access to various ontologies is also important. There are several already available, and as an example the W3C has published a document on representing WordNet in RDF/OWL. Research for this study could not find enough Web pages with semantic information relevant to arms transfers to be worth exploiting. Furthermore, there is little reason to suspect that entities preparing such documents are likely to spend the additional effort required to provide semantic markup. Techniques for accessing the web and locating relevant content vary widely depending on the goal. Many techniques have a very narrow focus that allows researchers to ignore complications that are out of scope. In this study, we limit the domain of consideration to weapons transfers; however, our method to accomplish this considers diverse content sources, which introduces several complications for analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

III. METHODOLOGY

A. PROGRAM DESCRIPTION

To evaluate our hypothesis that an automated approach to identifying weapons transfer information would be more efficient than manual methods, we designed a system to locate, acquire, and interpret weapons transfer content from the Web. The application was written in the Python programming language. Some modules that were used are provided with the Python distribution and are not covered here; others were provided by third parties including `httplib2`, `html5lib`, `NLTK`, and `MySQLdb`.

- `Httplib2` is a library that can perform several HTTP and HTTPS methods including GET and POST and is useful for retrieving Web pages and header information provided by the server such as the content type.
- `Html5lib` comes from Google's project hosting Web site, and provides functions to read a Web page and build a tree of the content; one option is the XML Document Object Model (DOM) tree structure. This module can also handle errors presented by improperly formatted HTML.
- `NLTK` (the Natural Language Toolkit) provides several functions for parsing text. In this program NLTK is used for WordNet access, and tests using sentence tokenization.
- `MySQLdb` is simply a library that can be used to perform functions against a MySQL database.

Figure 1 shows the system structure. The main program coordinates activities between three core components: Search, Parser, and Database. Generally, the main program retrieves a list of search terms from the database class, and feeds those search terms to the Search class. The Search class then goes to each of the desired search engines and retrieves a specific number of URLs for the search term. URLs are put into a database, and eventually the search engine downloads their content. Links discovered on each page are not added to the search queue as we assume the search engines will have

rated the relevancy of these links in their search results provided earlier. Content is parsed and statistics from the parsing are stored back into the database. The search class aggregates functions for each of the search engines. The database class regulates access to the database, and abstracts data from the SQL for inserting and retrieving into the database.

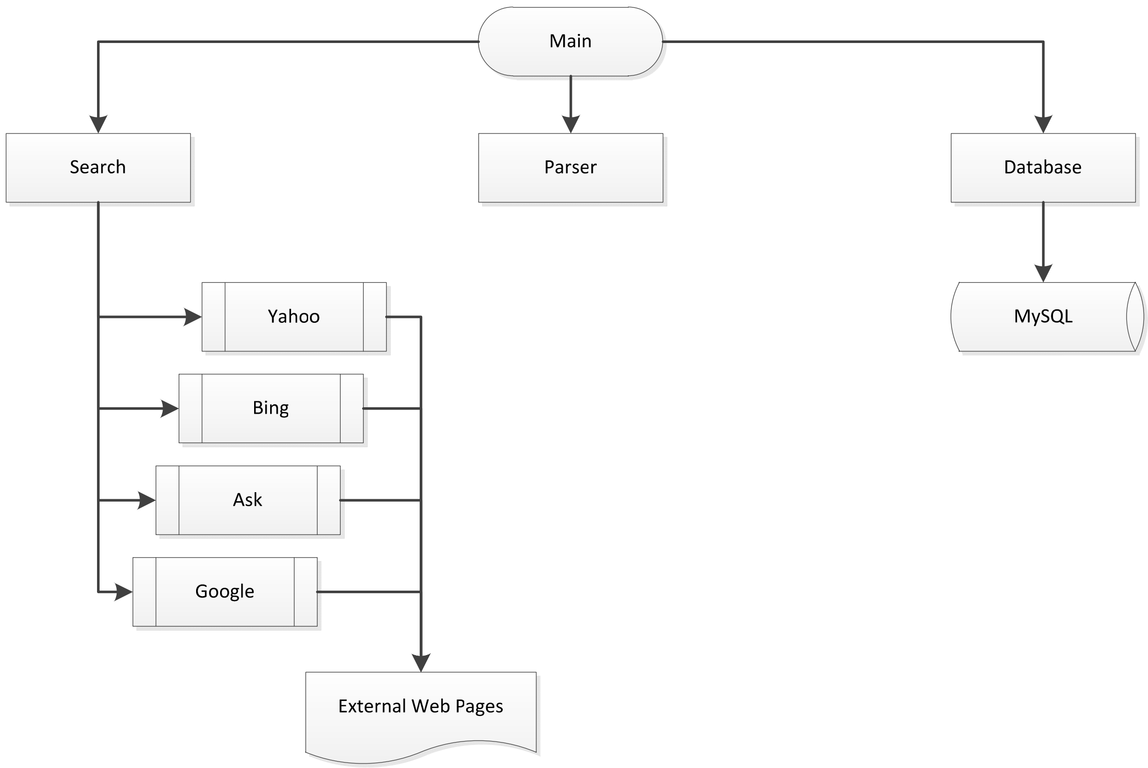

Figure 1. Program structure

B. STORAGE MODEL

Data for our program is stored in nine tables in a MySQL database. The tables are categorized into three groups: those used by the retrieval engine, those used by the parser, and one table used to collect statistics about execution of the program. A diagram of the tables and their relation is provided in Figures 2 and 3.

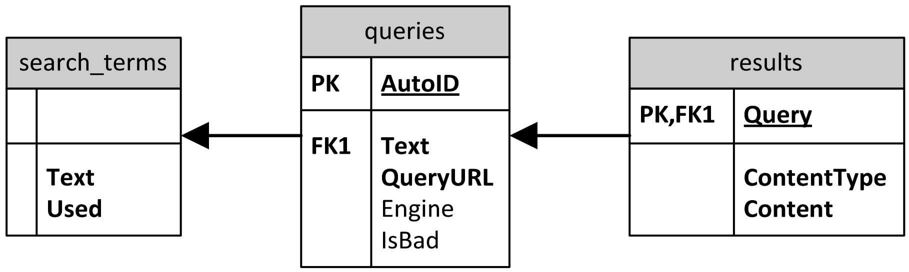

Figure 2. Page retrieval tables

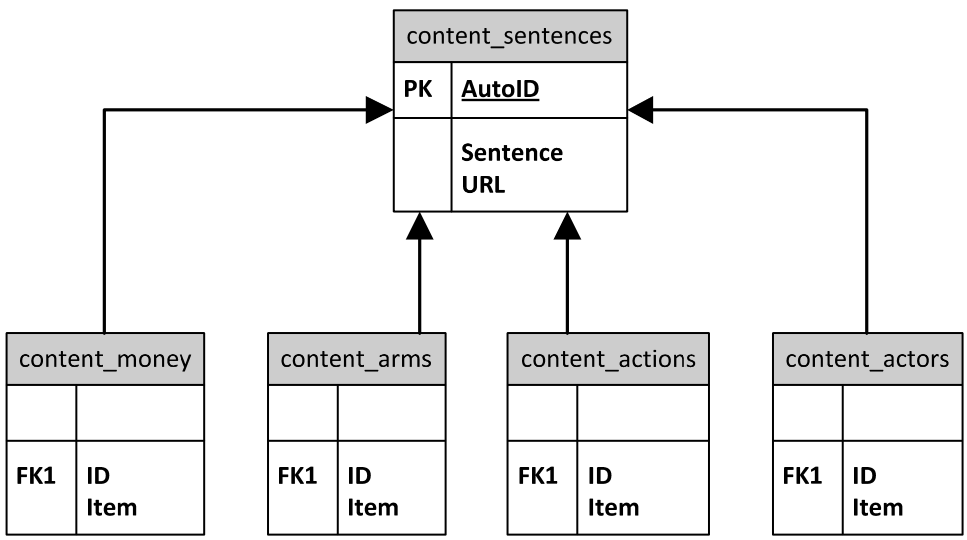

Figure 3. Parser tables

The page retrieval group has three tables. The `search_terms` table holds text which is input to the search engines. The `Used` column is a progress indicator to show that a search term has already been processed. This could assist if the program is run multiple times, or multiple crawlers are executing in parallel. The queries table holds the list of URLs retrieved from each search engine based on the query that was performed. The `Text` field maps back to the `Text` field of the `search_terms` table. The `QueryURL` is the URL retrieved from the search engine, the `Engine` is the actual search engine that was queried (e.g., Google, Yahoo) and the `IsBad` field is a Boolean,

which can be set if the URL cannot be reached (as when the domain may no longer resolve to an IP, or HTTP error codes such as 403 or 404 may have occurred.) The results table stores Web page content and the content type.

The parser tables shown in Figure 3 map features to sentences. This configuration allows each sentence to have multiple actors, actions, arms, and monetary units associated with it. For instance, a sentence indicating a weapons sale between two countries can have two entries in the content_actors table associated with one entry in the content_sentences table.

MySQLdb also contains methods to sanitize data that helps prevent accidental modification to, or exposure of, information in the database. This sanitization procedure adds quotation marks to certain characters which would be otherwise interpreted as SQL commands. All methods that incorporate parameters into queries have run each parameter through the escape() method provided by the MySQLdb module.

C. PAGE RETRIEVAL ENGINE

The crawler is written in Python and uses the httplib2 module. The first step is populating the search_term table with keyword terms to be supplied to existing search engines. In our project, so five basic phrases were used: “weapons sales,” “arms proliferation,” “gun control,” “global arms race,” and “weapon imports and exports to other countries.” Since the focus for this particular project was “state actors,” a list of country names was downloaded from the ISO [30].

Each country was paired with the word “weapons” and this phrase was added to the search_term table. This resulted in phrases such as “United States weapons,” and “Congo weapons.” Next, each search engine was queried for each phrase, and the search class attempted to fetch 100 URLs for each. Some search engines such as ask.com will present an inconsistent number of URL results per page instead opting to show sponsored results or links to images instead. URL results were extracted from the text of the Web pages returned. Each search engine had different query parameters, and different ways to indicate a search result as seen in Table 1.

Search Engine	Base Index	Index Increment	Anchor Pattern
Google	0	10	<h3 class=”r”>
Bing	1	10	<div class="sb_tlst"><h3>
Yahoo	1	10	<h3><a class="yschttl spt" href=
Ask	1	1	<a id="r\d+_t" href=

Table 1. Search engine properties

For the first page of results, Google does require a base index but results other than the first page require a GET variable, “start,” set to the correct index for the result increment desired. Also, Google requires that the requestor have a well-formed User-Agent string. It is unknown what exact forms Google accepts but it is assumed that this is a simple attempt to prevent automated programs from performing Web queries against its site. A sample user agent string was retrieved from Chrome in Linux and set as the default user agent for all search engine queries performed by the httplib2 module:

> Mozilla/5.0 (X11; U; Linux x86_64; en-US) AppleWebKit/534.16 (KHTML, like Gecko) Ubuntu/10.10 Chromium/10.0.648.127 Chrome/10.0.648.127 Safari/534.16

Yahoo search results were odd due to the format of their links. For example, while searching for “test” the following shows up in the Web page source:

> <h3><a class="yschttl spt" behref="http://search.yahoo.com/r/_ylt=A0oG7hUrkfVNRgkAnvlXNyoA;_ylu=X3oDMTEyNmRiMTNmBHNlYwNzcgRwb3MDMQRjb2xvA2FjMgR2dGlkA0RGRDVfOTE-/SIG=114nepfn3/EXP=1307960715/**http%3a//test.com/" data-bk="5055.1">

The link that the user sees on the Web page for this is http://test.com. Here, we must strip out everything before the correct URL that is demarked by two “*” characters.

Next, it is necessary to convert hex representations of characters to their ASCII equivalents (e.g., “%3a” to “:”, “%23” to “#”). Similar conversions were added for “&”, “(“, “)”, “+”, and “?”.

The Ask.com search engine adds a numeric variable to each search engine result. It also uses regular expressions like “\d+” to represent sets of characters. This injects some variation into the result set. For example, when searching for “test,” the first three results on the page have the following id classifications:

<a id="r1_t" href=”http://test.com/”

<a id="r1_t" href="http://www.speakeasy.net/speedtest/"

<a id="r2_t" href="http://www.ask.com/wiki/Test"

While searching Ask.com’s Web page source file, the “id” value did not correlate with the result number. A regular expression matching any number was used because we do not care which number is in the id field, we simply wish to grab the URL following it.

Besides using Google, Yahoo, Bing, and Ask.com, this program will initiate DNS queries for URLs retrieved from those search engines, and will make full TCP connections to the resulting IPs. We executed the server in Amazon’s cloud hosting environment [31], which was used to actually make the connections to the URLS and retrieve their contents.

D. PARSER

Here we describe the techniques used to parse Web pages retrieved by the crawler. The goal is to get the content from Web pages into a common format, then search that content for meaningful combinations of features. We restructure information from each Web page into a sequence of sentences, and then search for predefined weapons, actors, actions, and money. Each feature is specified using keywords or in the case of money, a grammar.

1. Web Page Interpretation

Web pages are commonly formatted in HTML. The Html5lib python library was used to parse documents into DOM trees. Only the content that was identified between <body> </body> tags was inspected. These trees were well-formed due to error handling provided by the module. Missing tags were replaced in their most logical location if that could be determined. From this point, the nodes in the DOM tree were looped over recursively and the content was appended together based on certain rules. Table 2 shows tags which likely indicate breaks in content. Tags that do not indicate a logical break in content include <b> and <font>.

Content Dividing Tags
<span>
<div>
<table>
<tr>
<td>
<h>
<script>

Table 2. Tags that are likely to represent a logical break in the content of the HTML file

The content between the tags in Table 2 was appended together with a period in between to form separate sentences. The content between the other tags were simply appended together, with a space. Using the DOM tree parser, it was unnecessary to remove the excess whitespace found in many HTML files.

At this point the Web page has been converted into a group of sentences. These sentences may have odd formatting due to how the parser appended sentences together.

It is a difficult task to build a parser that will handle every single way content might be formatted on the Web. Menu systems, for example, can be defined using <div> and <span> tags, which result in short sentences using this method. But this is acceptable for a menu system as it is not expected to contain the group of features we hope to find. The sentences were then tokenized into a list of words. NLTK contains tokenizers that can split content up according to several patterns. The Punkt tokenizer was tested; however, without training it fails to tokenize properly. We developed a custom tokenizer to accomplish this task.

According to the NLTK developers, "Tokenization turns out to be a far more difficult task than you might have expected. No single solution works well across-the-board, and we must decide what counts as a token depending on the application domain" [27]. For instance, periods do not necessarily indicate the end of a sentence, nor do question marks or exclamation points. There are many cases where an abbreviation, number, or some other form of text may contain any one of these punctuation marks. Examples include the initials for a person's name (e.g., "J. Smith" or "J.S."), money (e.g., "$100.00" or "$1.00,00"), abbreviations such as "assoc." for "association," parenthetical notations, and errors converting from one encoding to another. The latter two cases were not handled in the design of this custom tokenizer. Question marks and exclamation points are assumed to end a question or sentence, respectively, and the algorithm below attempts to identify whether periods correctly end a sentence or not:

1. *Create a list by splitting data on occurrences of [.?!]*
2. *Set previous match conditionals to False.*
3. *For each item in this list:*

    3.a. If length(item) < 3, pop the list, and concatenate the two items together with a period in-between. Set this new string as the current item, and set previous match conditionals to False.

    3.b. If the previous match conditional is True, or if the previous match end number conditional is True and the current item starts with a number, pop the list and add a period. Store the result in a temporary variable.

3.c. Concatenate the temporary variable and current list item, and push the result onto the list.

3.d If the result that was just added to the list contains an exclusion pattern, set the previous match conditional to True. If the result ends with a number, set the previous match end number conditional to True. Reset any conditionals to False, which were not explicitly set to True.

The original selection of exclusion patterns and method of using them were provided in [32] and are listed in the appendix. The exclusion pattern also matches '(?: |\.)[a-zA-Z]$", which will be true if the end of a sentence has a space or period followed by a single letter and helps identify things like initials.

Python regular expressions are not Perl-compatible [33]. An example is that more complex regular expression techniques such as look-behind assertions will not work with more than three characters in Python. It was not necessary to perform look-behind assertions with the re module provided with Python, but this restriction did affect how this algorithm was implemented.

2. Feature Extraction

The main goal of the parser is to scan each sentence for known keywords. We chose a test application of arms sales. For this application, keywords are in four categories: weapons, actors, actions, and money. Each group has unique requirements that guide how they are chosen. They were identified inside of sentences using regular expressions.

a. Weapons

An initial list of weapons was acquired using the NLTK WordNet interface. Select hyponyms from the original word "weapon" were obtained and an exhaustive search was performed on the resulting list for those weapons. The base hyponyms were "light arm," "weapon of mass destruction," "missile," and "gun." Some weapon types were not included because they were not identified as significant items between state actors like brass knuckles, swords, pellets, pikes, slashers, bows and

arrows, etc. Some common weapons may be omitted from the hyponym selection, some names are too specific, and some may simply be formatted in a manner different to that which is written on the Web. For example, a hyponym of "missile" is "heat-seeking missile," but it is rare that a Web page has the weapon formatted exactly this way. Other possibilities include "heat seeking," "infrared," or "IR." A misspelling may also trigger a false negative; these can often be found by searching for all misspellings of the weapon systems of interest. We did not implement such a system

False positives can occur. For example, one weapon is a SAM, or Surface-to-Air Missile. A case-insensitive regular expression search for "sam" will net a wide list of results of people named "Sam," or partial matches on words like "same." Techniques to guard against these situations include restricting the common characters surrounding the weapon names of interest. Words surrounded by symbols or white space have a higher likelihood of being correctly identified. A list of weapon names is included in the appendix.

b. Actors

In this study, we chose to focus on state actors. The ISO country list is provided as an XML formatted file and contains official country names, some of which are not common. Table 3 shows some examples. If a country name had a comma in it, the part before the comma was kept in our table. Also, countries may be referred to differently in other languages or other parts of the world, so we had to add entries for them. Appropriate additions were also made (e.g., "Russia").

ISO Name	Unofficial Name
CONGO, THE DEMOCRATIC REPUBLIC OF THE	Congo
IRAN, ISLAMIC REPUBLIC OF	Iran
KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF	North Korea
KOREA, REPUBLIC OF	South Korea
LIBYAN ARAB JAMAHIRIYA	Libya
RUSSIAN FEDERATION	Russia

Table 3. ISO official country names compared to common country names

The correct identification of state actors has many of the same possibilities for false negatives as for weapons. A Web page may simply have misspelled a country's name, or it may be spelled differently due to being listed in a different dialect or language. For instance, Egypt is also referred to as Misr in Arabic, which may also contain an accent mark. Another issue preventing correct state-actor identification is classifying specific agencies or organizations as references to an entire country. As an example authors may refer to Washington, the Pentagon, the White House, or another government organization when reporting news related to the United States.

Proper entity identification requires the collection of aliases for those entities. No exhaustive list providing aliases for countries was readily available so this component was approximated by adding a limited list of well-known alternate country names.

Once an entity is identified, future references to that entity may take the form of a personal pronoun such as "they," "them," "we," "it," etc. and may become increasingly confusing with multiple entities being referred to in such a manner. This association would be important; however, the method for accomplishing this task is not the focus of this study.

c. Actions

Several wordings may indicate a transfer of weapons from one state to another. In weapon sales, there may be no currency involved in the transaction, or items could have been traded or donated. There may be many ways the transfer is described, so the list of verbs should be wide enough to capture common methods. Table 4 shows verbs that were searched for. Each of these is inserted in a regular expression to identify matches. Ideally a parser that can analyze stems and affixes of key verbs may be preferable to one individually searching for the each of the variations.

Transfer Verbs
"bought", "buy", "buying", "sold", "sell", "selling", "sale", "acquire", "acquired", "export", "exported", "import", "imported", “importing”, "purchase", “purchased", "gave", "give", "given", "take", "took", "taken", "received", "receive", "distribute", "distributed", "traffic", "trafficked", "trafficking", "barter", "auction", "shop", "shopped", "shopping", "procure", "procured", "procuring"

Table 4. Verbs that could be used to describe a transfer of weapons

d. Money

With monetary values within text our focus was on numeric representations, where word-based representations such as “one hundred thousand dollars” are not considered. To distinguish money from other numeric text (e.g., version numbers), we required two conditions be met. There must be a currency symbol followed by a properly formatted number (e.g., $1,000), or there must be a properly formatted number followed by a currency name or abbreviation (e.g., 1,000 dollars). This is because the absence of a currency symbol, name, or abbreviation greatly reduces the possibility that the number refers to money. Commas or decimals may or may not be

used in monetary amounts, or the order of commas and periods within currency may be switched. Given these rules, we constructed a grammar to use for money detection, which is shown in Table 5.

<table>
<tr><th>Money Grammar</th></tr>
<tr><td><Money> → < prefix> <number><postfix> | <prefix><number> | <number><postfix></td></tr>
<tr><td><prefix> → <symbol> | <abbreviation></td></tr>
<tr><td><number> → <number body><amount> | <number body></td></tr>
<tr><td><number body> → <digits><dotted decimal> | <digits><comma decimal> | <digits with commas><dotted decimal> | <digits with dots><comma decimal> | <digits with commas> | <digits with dots> | <digits></td></tr>
<tr><td><postfix> → <abbreviation> | <currency name></td></tr>
<tr><td><digits> → [1-9]\d*</td></tr>
<tr><td><digits with commas> → [1-9]\d{0,2}(,\d{3})+</td></tr>
<tr><td><digits with dots> → [1-9]\d{0,2}(\.\d{3})+</td></tr>
<tr><td><dotted decimal> → \.\d{2}</td></tr>
<tr><td><comma decimal> → ,\d{2}</td></tr>
<tr><td><symbol> → $ | € | £</td></tr>
<tr><td><abbreviation> → USD | EU | EUR | GBP</td></tr>
<tr><td><currency name> → dollars? | euros? | pounds?</td></tr>
<tr><td><amount> → thousand | hundred thousand | million | billion | trillion | k | m | b | t | mn | bn | tn | mln | bln | tln | mil | bil | tril</td></tr>
</table>

Table 5. A grammar to describe possible money formats

The grammar listed in Table 5 matches multiple representations of American, European, and British currency, while also matching mixed currency formats.

An example of a mixed format would be $1,000.00 GBP. This grammar was implemented with a hierarchy of regular expressions in Python. Adding new currencies to this grammar is also relatively easy; for instance, if we wanted to add the Canadian dollar, we would add "C$" to the list of symbol terminals, and "CAD" to the list of abbreviations. Currencies with alternate numerical formatting would need to be accounted for within the <number body> non-terminal and subsequent terminal conditions.

3. Content Search

Once the Web pages were properly tokenized we searched for features within each of the Sentences. Sentences were analyzed to see which combinations of features resulted in the most accurate matches. Upon finding a match, the content_sentences table was populated with a sentence and the URL of the corresponding Web page. Features identified in the sentence were stored in the remaining content tables based on type (e.g., actor, action, weapon, or money), with an identifier that linked the feature to the original sentence. This allowed multiple features in each category to be recorded for a single sentence. The parser ignored sentences with fewer than two matching categories of features. Too many false positives were encountered with only one matching feature category. Sentences with only two matching feature categories still had a high chance of being false positives; however, certain combinations may be more accurate than others. For instance, a combination of money and action features might result in a high rate of false positives, but the presence of weapon and money figures may not.

E. EXECUTION

Upon executing the program, 2,144 pages were downloaded using Google's custom search applications programming interface (API). Each page was broken into sentences and then parsed for weapons, money, actor, and action features and then stored in a relational database. A random sample of 100 pages was also selected for precision and recall analysis. The machine configuration used to parse through the results included a dual-core Xeon processor running at 3.2GHz, with 4GB RAM running Linux.

F. SEARCH ACCURACY

There are several aspects of our data retrieval methodology where accuracy was affected. First, the retrieval of search results from Google is based upon search terms created for this study. The search terms used may not have yielded the most relevant content from Google's indexes. Second, Google's indexes are not comprehensive and so even with the most targeted search query possible, some valuable results from the web may not have been returned. Third, the number of results obtained for each query was limited to 10 resulting in about 2400 actual search results returned with duplicates excluded. The random sample of 100 pages was chosen from the set of 2400 results. Last, the number of terms used to identify features, and the limited ability to handle grammatical variations affected how successfully sentences were matched for relevant content. No attempt was made to identify and exclude negations, factually incorrect information, or special emphasis that may imply an alternate meaning in the form of humor, or sarcasm. Also, sentences containing multiple features within each category were not decomposed to classify the actions of each individual actor, or associate the other features with them.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. RESULTS

Here, we show that automated techniques for retrieving and parsing information to identify weapons transfers can be more efficient than manual methods. To show this, we analyze the type of content discovered on the web, the amount of time it took to analyze it, the precision and recall of our technique, variations in the content that made parsing difficult, and several samples of the data uncovered from the results.

A. PAGE STATISTICS

1. Content Encoding

We checked the encodings for each page downloaded using the crawler. If the character set was not specified using an HTTP header, we would attempt to decode using several popular standards such as UTF-8, ISO-8859-1, etc. Where the encoding could not be determined, we would record "UTF-8" as the character set. 28 sites that specified a UTF-8 character set (1.54%) were not encoded properly; no other pages generated errors.

The encodings from the entire data set are shown in Table 6.

Encoding	Count
UTF-8	1818
ISO-8859-1	311
Latin1	6
ISO-8859-15	2
Windows-1251	2
Windows-1253	1
Windows-1255	2
US-ASCII	2

Table 6. Detected encodings

2. Performance

The processing time of the program while parsing a random sample of 100 sites is shown in Figure 4 plotted against the page length. The number of sentences is plotted against the page length in Figure 5, illustrating a very similar relationship.

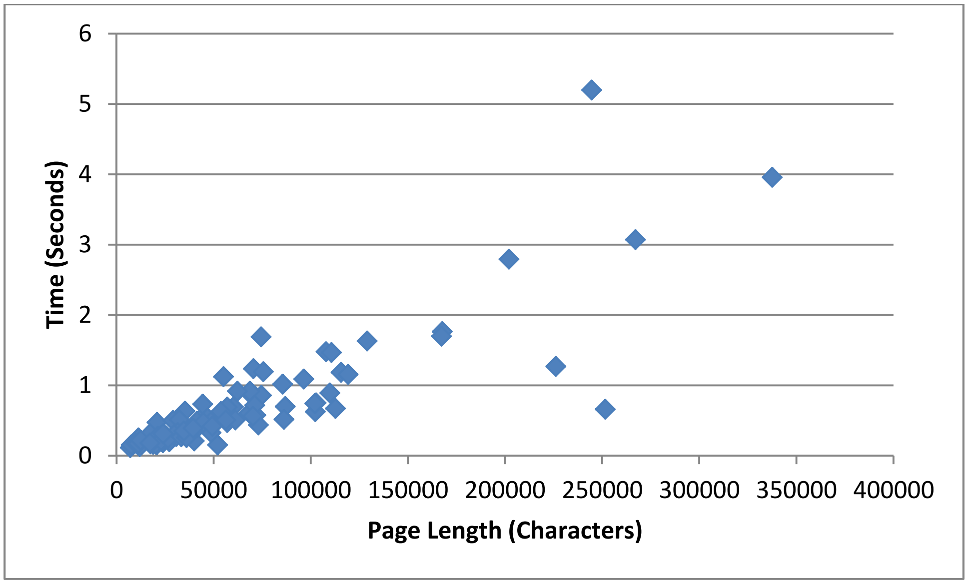

Figure 4. Total parse time compared to page length

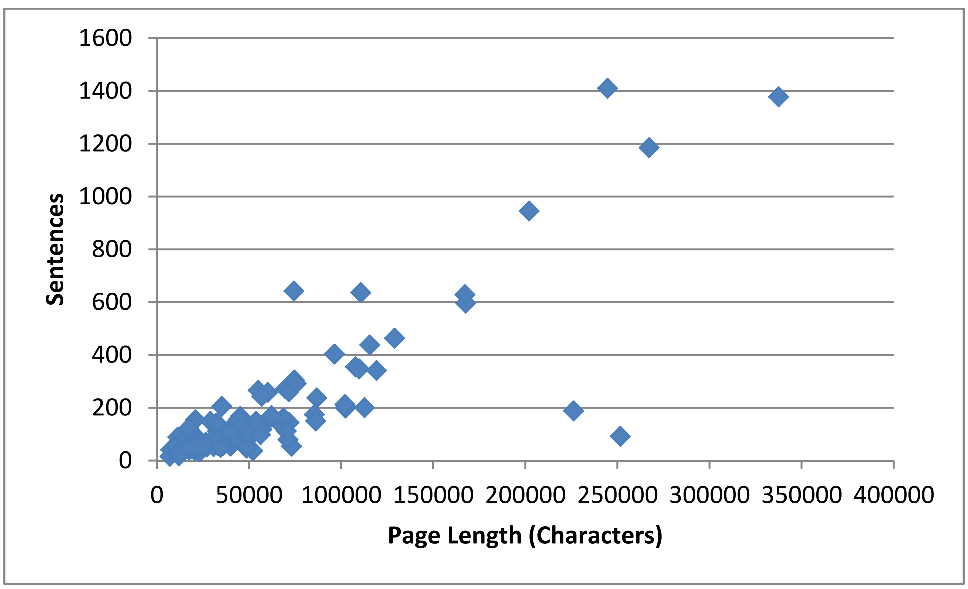

Figure 5. Number of sentences compared to page length

Most pages were under 125,000 characters, and took under two seconds to parse. Figure 4 shows a few outliers, indicating short or long parsing times compared to page lengths. Factors such as disk I/O speeds and process context switching within Linux introduce expected delays but they should affect all results, and do not likely account for the outliers. Figure 5 shows that a significant factor affecting the speed of these operations is the number of sentences on each page.

As an example, two pages with a length between 225,000 and 250,000 have 200 or fewer sentences, and this accounts for the shorter parsing time.

3. Feature Statistics

We focused on sentences whose words matched three and four features of money, arms, actors, and actions. Table 7 shows the number of sentences from both the limited result set and all results that contained matches for three or four of the feature sets. There were approximately 1225 words per page on average, yielding about 2,627,625 words across the corpus.

	100 Pages	**All Results**
Three Features	148	3160
Four Features	6	338

Table 7. The number of sentences that contain three or four feature set matches

Breakdowns of the combination of feature set matches in sentences are provided in Figures 6 and 7.

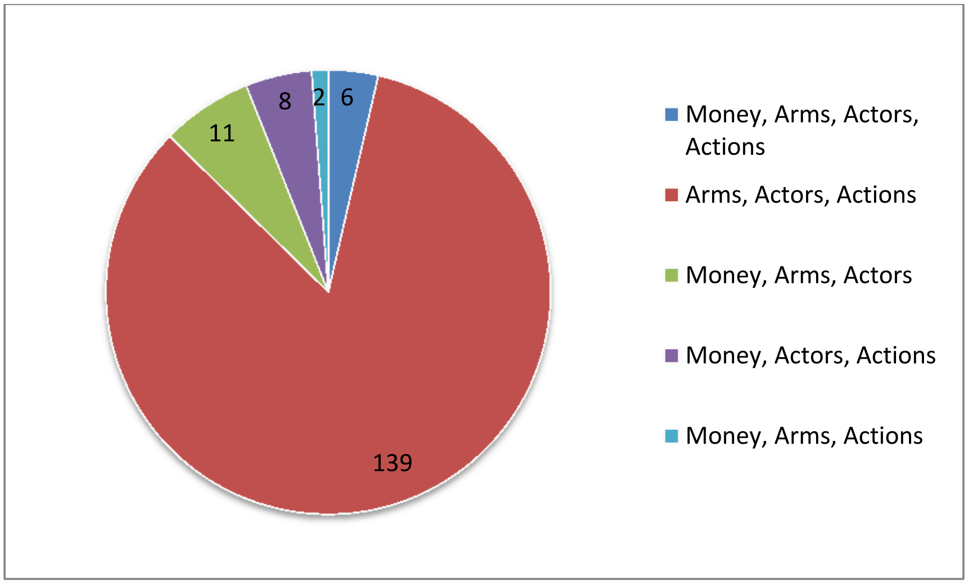

Figure 6. Feature set matches for 100 pages of results

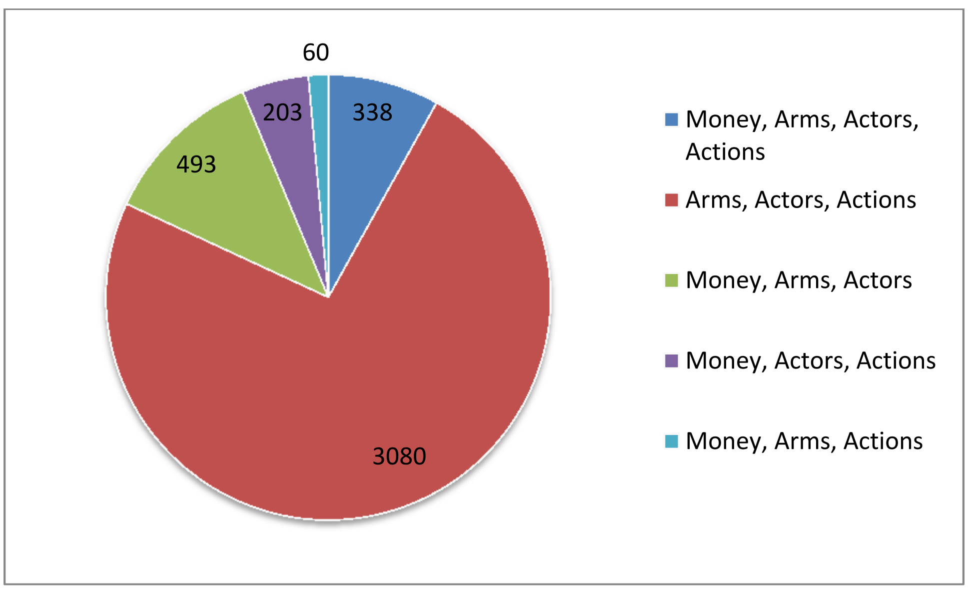

Figure 7. Feature set matches for all results

The most common feature set group was Arms, Actors, and Actions, while the Money, Arms and Actions feature set had the smallest percentage.

4. Precision and Recall

Precision and recall of our retrieval of relevant sentences were measured according to several criteria. If a sentence did not describe some sort of weapons transfer, it was counted as a "negative." Sentences had to include items in the original list of actors, actions, or arms to count as "positive," which for instance excluded sentences with "artillery shells" because it was not in the original list of weapons. It also excluded sentences containing "Dubai," a city in the United Arab Emirates (UAE) since this is not a country. Feature matches generally had to be for the whole word.

However, for evaluation any misspellings on pages that resulted in a feature not being correctly identified were not counted negatives for this reason alone. For example "Syrai" counted as a match to "Syria." Partial matches for features that yielded the correct understanding but not a specific detail or variation on the original word were allowed. Examples include matching "ballistic missile" to "intercontinental ballistic missile," or "Congo" to "Congolese."

A total score for a sentence match was calculated by dividing the number of features identified by the number of correct feature matches for a sentence. For example, the sentence "The United States sold Britain $1 million dollars of weapons." has five features, which are "United States," "sold," "Britain," "$1 million," and "weapons," and the score would be 1 if all features were correctly identified. If one of the features differed, the score would be 4/5 or 0.8.

Examples of sentences that did not describe weapons transfers and could be false positives included those detailing laws related to weapons activity, media reports on weapons testing, and weapons related items such as purses designed to conceal handguns. Not all sentences that did describe weapons were useful. For instance, several sentences were found on a bulletin-board system in which people debated the history of an armed conflict between two countries. Some sentences about arms sales were facetious in nature. There were also sentences that confirmed that there were in fact no weapons transfers.

Speculation can help establish a link between two or more actors, but could be based on factually incorrect information and can detract from accurate statistics on actual arms sales. It was defined as sentences describing possible outcomes of past, present, or future scenarios. Speculative sentences were found by the parser and were counted if they described weapon transfers. As an example, a news page may offer suggestions to explain certain activities, but they may not be based on factual information. Not all weapons transfers are publicly documented, and reading a news source that explains weapons "may have" come from a certain location do help establish links between countries but should not be treated as factual. Tables 8 and 9 detail the precision of the parser results for whole sentences and the count of individual features identified in each of the sentences, respectively.

Feature Sets	**False Positives**	**True Positives**	**Precision**
Actors, Actions, Arms, Money	1	5	0.833333
Arms, Actors, Actions	91	42	0.315789
Money, Actors, Actions	3	5	0.625
Money, Arms, Actors	3	2	0.4
Money, Arms, Actions	0	2	1

Table 8. Precision metrics by sentence

Feature Sets	**False Positives**	**True Positives**	**Precision**
Actors, Actions, Arms, Money	1.8	4.2	0.7
Arms, Actors, Actions	98.659	34.341	0.2582
Money, Actors, Actions	4.05	3.95	0.4942
Money, Arms, Actors	3.25	1.75	0.35
Money, Arms, Actions	1	1	0.5

Table 9. Precision metrics by feature counts

In Table 8, the best performance occurred with the money, arms, and actions group; however, Table 9 shows that accounting for individual features yields a more granular precision metric and shows that the actors, actions, arms, and money group actually performed better. For the money, arms, and actions group, both of the sentences correctly matched half of the features, which corresponds to the precision metric from

Table 9. The arms, actors, and actions group had a wide disparity, meaning that it is likely to trigger many false positives for unrelated sentences. The money, arms, actors, and actions group had better overall performance identifying relevant sentences 83% of the time, and associated features correctly 70% of the time, where the feature comparison gives a more accurate representation of the parser performance.

The "Actors" group had more than twice as many missed elements as any other feature set. Example terms on pages that could not be matched from our search terms included "other middle eastern states," "Burma," "PAIGC," and "USSR." Often, a person's name was used to represent the actions of a larger group, as for instance the prime minister of a country to imply the country itself. Actions that were missed included but were not limited to words such as "shipment," "supply," "supplied," and "supplies." Our parser relied on exact matches but could benefit from word-sense analysis and stem and affix matching (morphology).

The arms feature set had weaknesses on both general and specific items. Many sentences would only state "weapons" or "small arms"; however, some sentences would enumerate specific weapons such as "bazookas." Some descriptions were vague enough where it was difficult to discern if the author was talking only about weapons. An example is "nuclear and missile technology"; without more context, "nuclear" could refer to non-weapons technology such as nuclear-power generation. Feature matches for money also encountered several anomalous situations. Examples such as "US$78,34312" would be interpreted as "$78,343" because the original format was not part of the grammar. It could be that the original value was supposed to be $7,834,312, $78,343.12, etc. Currency could be misidentified, as for example "2007EU," which does match the grammar, but refers to a set of reports issued by the European Union.

Recall statistics were difficult to estimate. Factors that affect the accuracy of this measurement include the limited data available on the Internet, and the lack of a control group or authoritative source on weapons transfers to compare against. Small-scale comparisons were made against two different sources, the Stockholm International Peace Research Institute (SIPRI) [34], and results directly from Google. The SIPRI database lists the value of imports and exports between countries, and weapon categories (e.g.,

aircraft, artillery) by year. Our findings were compared to the imports and exports of 10 countries listed in SIPRI between 1999–2010 specifically to see if the recipients and supplier countries were also found. The 10 countries chosen were Afghanistan, Albania, Algeria, Angola, Argentina, Canada, Egypt, Greece, Iran, and North Korea.

Results are shown in Table 10. The “Imports From” and “Exports To” columns list the number of other countries that the country in question either imported from or exported to. Using our data set of 100 Web sites, all sentences referencing each of the 10 countries were compiled. If another country was cited as the recipient of or supplier to one of the 10 countries, and a relationship between the two countries was already in the SIPRI database, it was counted under the “Found in SIPRI” column. If a country was identified that was not listed as a recipient or supplier, we counted it under the “Not in SIPRI” column. The recall for these 10 countries ended up being about 6/(59+96) or about 3.87%.

There are several reasons why this percentage is small. First, our data set of 100 Web pages was randomly selected and may not have included information on each of the 10 countries. Second, SIPRI may include information available from sources that cannot be reached by a Web crawler or are not documented anywhere else. Transfers that were found by our program but were not listed in the SIPRI database were limited but help to establish links between countries. In the case of Albania, there was one Web page that contained a sentence describing how Turkey denied an arms shipment destined for Armenia. For Iran, one Web page indicated that they were exporting weapons to Congo, another indicated that it imported weapons from the UAE, and another indicated a possible trafficking network including Venezuela and Uruguay, none of which were in the database. There was also a sentence where North Korea denied that they were importing uranium from Congo for making atomic bombs. Though this is not a confirmation of a weapon transfer, the relationship between the DPRK and Congo may be useful to know.

Country	Imports From	Exports To	Found In SIPRI	Not In SIPRI
Afghanistan	10	0	0	0
Albania	4	0	0	1
Algeria	13	0	0	0
Angola	12	1	0	0
Argentina	10	3	0	0
Canada	14	43	0	0
Egypt	10	0	0	0
Greece	14	4	0	0
Iran	7	3	3	4
North Korea	2	5	3	1
Total	**96**	**59**	**6**	**6**

Table 10. Findings matched from 10 countries in the SIPRI database

A different recall statistic was calculated for the 100 Web pages that compared the transfers identified by the parser to those that should have been identified, but were not. Several difficult cases were identified. One involved interpreting tabular results for arms imports and exports by weapon type and country. If there were no features that would give a timeline and no way to determine how they were separate except by country of origin, or possibly weapon type, one transfer was recorded for each country. Another scenario included a transfer being described over multiple sentences, which we counted as one transfer. There were also issues with possible duplicate information due to the overlapping of general and specific claims. As an example, the parser flagged one sentence as a transfer that described an approximate dollar amount of weapons exported every year, and flagged another sentence that described a specific transfer to another country. In this case, both were counted as transfers as it is possible that the former accounts for other weapon transfers that the latter does not. Descriptions of potential future weapons transfers, as well as weapons testing were also counted as transfers.

Comparing the number of transfers found by the parser to the number of transfers which should have been found overall in the data set, the recall rate was estimated to be approximately 34.5%.

False positives occurred when the parser failed to properly distinguish separate sentences within Web pages, which led to more feature matches. They also occurred when searching for fewer features per sentence. False negatives occurred when we failed to match key features because they were not in our feature list. Examples include:

1. *"This sale was the first Iranian export of the domestically produced version of the missile, and marked the first instance in which Iran exported complete missile systems"* [35]

2. *"In the Czech Republic, only licensed gun owners (for certain arms)*[3] *[footnote in original] may lawfully acquire, possess or transfer a firearm or ammunition" [36]*

3. *"Documents issued by the Latvian export control authority in 2007EU Annual Reports on Export of Arms and other Military GoodsExport control regimesThe Australia Groupis an informal international forum, having the objective of preventing the spread of chemical and biological weapons" [37]*

4. *"In 1997 the Tanzanian government initiated a task force to curb the smuggling of small arms into the country from the Democratic Republic of the Congo (DRC) and Zambia"* [38]

The first sentence is classified as a true positive, as within the context of the original page it does indicate the actual export of a weapon. The second sentence is classified as a false positive because it describes a law relating to weapons, not an actual weapons transfer. This sentence matched three feature categories for arms, actors, and actions. These were the most common feature categories found together and also had the highest rate of false positives. The third sentence represents a false positive due to a parser error. Four feature categories were matched, but the HTML indicates that there were logical breaks in content that we did not include when deconstructing Web pages, which include the <p></p> and <br> tags. If the parser worked correctly in this scenario,

there would actually be four sentences, each ending at the words “2007,” “Goods,” “regimes,” and “weapons.” Generally the sentence boundary words are merged together without a space, except in the case of “groupis.” The reason the words “group” and “is” are not separated in the third sentence is due to the author of the Web site utilizing “ ” to indicate a non-breaking space. The fact that this space was not converted did not affect our parsing operation in this example. The fourth sentence is an example of a false negative. Though this does describe a transfer, “small arms” and “smuggling” were not in our feature lists, so our parser only matched three countries.

5. Data Visualization

We took sentences that matched four feature categories and identified at most two Actors and plotted the results as a graph. When exactly two actors were found, they were treated as linked nodes. A portion of the graph derived from the 2,144 Web pages is shown in Figure 8.

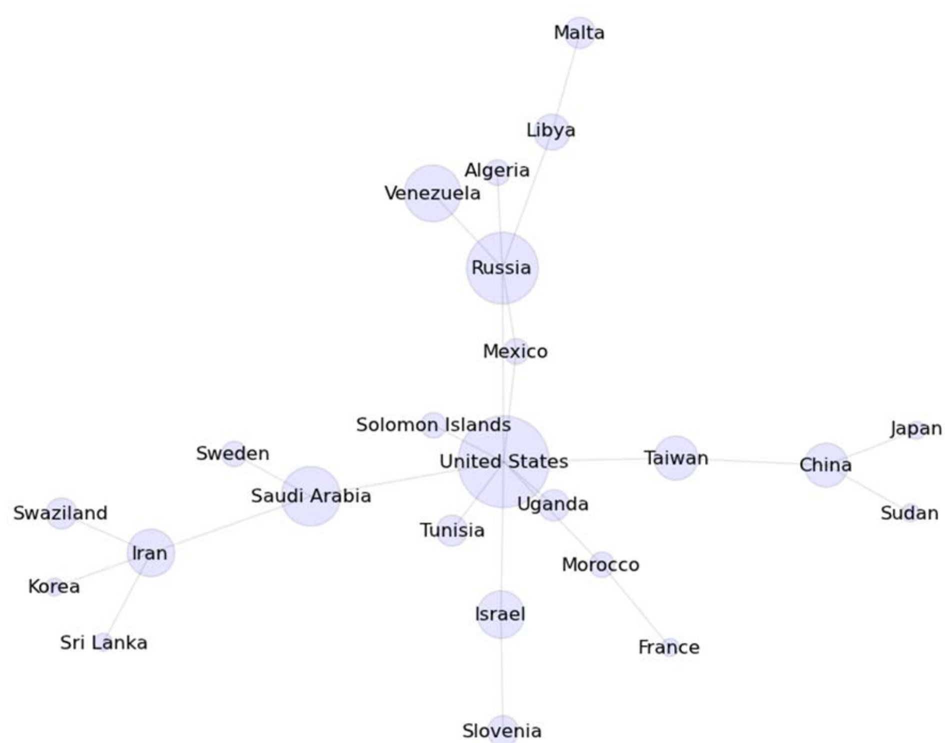

Figure 8. Graph showing countries that were mentioned with one other country in the same sentence

Figure 8 does not show evidence of weapons transfers, but does show countries that were mentioned together in a context that is likely about weapons transfers. Not all relationships are shown because this was generated from a limited data set, and countries that were not linked to others were excluded. Some relationships may be incorrect due to reasons described earlier for precision and recall. For example, the United States and Taiwan were referenced in one sentence, while Taiwan and China were referenced in another but their relationships are different. Some links may indicate a relationship that includes weapons transfers, and some indicate that a comparison between countries was made. All links represent the fact that two countries appeared in the same sentence one or more times. The size of the nodes represents the number of references to each country, and is independent of the number of links to other countries. With more data and a more advanced parser, a directed graph could be generated that may show actual weapon flows between countries.

Next, we drew a graph to account for sentences that referenced up to four countries. Figure 9 shows some relationships that were not in Figure 8. For example, there is now a link between Morocco and Venezuela. Saudi Arabia and Libya are not connected in the graph, nor are Japan and Malta; the node placement was a function of the graphing software.

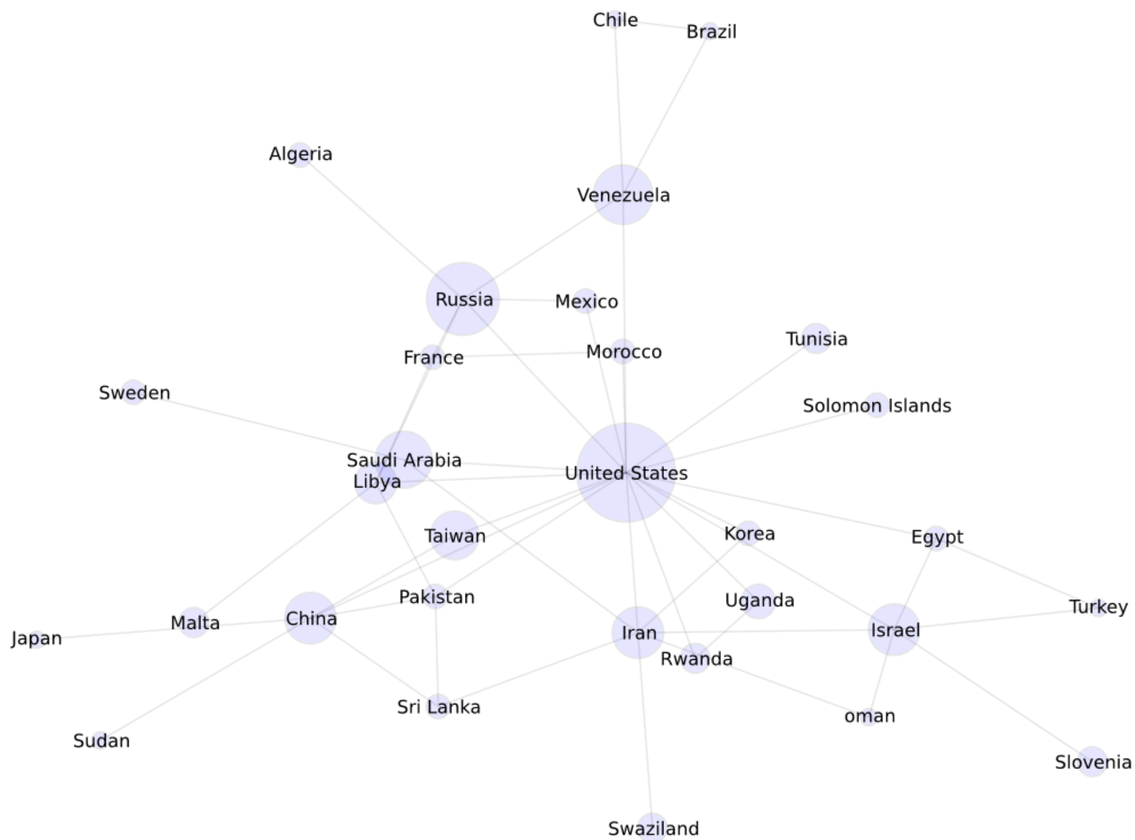

Figure 9. Graph showing countries that were mentioned with up to four other countries in the same sentence

Both graphs show the relative number of references to each country, which is denoted by the size of each node. More references may mean more supporting evidence of weapons transfers. Most were for the United States, while several references were found for Russia, Saudi Arabia, and Venezuela. Because both figures only show connected nodes, there may have been other individual countries with large numbers of references that were not displayed.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION

If relevant, current, and factual information could be successfully aggregated, it would benefit law enforcement agencies and policy makers, as well as contribute to other weapons proliferation research. We have shown a method of aggregating weapons transfer data from content available on the Web, and have demonstrated that this technique can identify relevant transfer information correctly 70% of the time among all sentences matching four feature sets. On average, it took .68 seconds to analyze individual pages and perform feature identification when executed against a random sample. This supports our hypothesis that automated mechanisms can improve the speed at which pages are analyzed for relevant content. We developed a retrieval system for acquiring relevant pages, a storage model for this data, and also a grammar to detect currency, which can be easily modified to identify new formats. This analysis was accomplished by querying search engines for relevant Web pages, extracting content from the pages, tokenizing the content into sentences and then searching the sentences for relevant features. Precision was measured without training data, and recall was measured using the SIPRI database. During analysis, we found that parsing content from the Web is difficult due to the level of variance in structure and format of available data. Though we were able to successfully identify features that we were specifically looking for, we were unable to handle the identification and tracking of related features. Due to the level of variance of Web data, an approach that can evaluate and incorporate unknown named entities into the parsing process would be ideal.

Future work can improve on this study in several ways, including identifying misspelled words and tracking subjects across multiple sentences. By interpreting common typographical errors correctly, and being able to correctly map certain pronouns to one or more candidate antecedents, these would increase the accuracy of the results. Identifying the correct version of a misspelled word automatically has been performed with 97–98% accuracy using techniques such as parts-of-speech tagging, and other forms of contextual analysis [39]. Another improvement would be identifying relationships between two named entities. As an example, when parsing an article referencing the

White House, there was no relation between it and the United States. Subject tracking across multiple sentences would require not only the ability to discover relations between known and unknown entities, but also to associate personal pronouns with those entities correctly. These tasks may make the parser more accurate but were considered outside the scope of this study.

Additional ideas for future work include detecting the direction of weapon transfers between two or more entities, which would enable the modeling of weapon flows. Another would be determining the dates and times of transfers that would be useful in determining relevancy of a match, and could facilitate the analysis of time-based trends. If exact date and time information cannot be inferred, an estimate as to whether the transfer is current could also be useful.

Web content can also contain information that is either wholly or partially incorrect. Therefore, a method to estimate the accuracy of information would also be a logical extension of this work. Web sites can be rated on their credibility based on both reputation and consistency of statements with authoritative sources. It could matter whether a sentence is in first, second, or third person, or whether the sentence is from a user comment uploaded to a Web site or other bulletin board system. Another factor is the level of certainty used to describe transfers; a statement would be less credible if the words "may" or "might" appear in it.

This work can be extended for use in other industries as well. In the business world, these data mining techniques could be used to identify relationships and agreements between companies, including international deals. In this scenario, the companies would be actors, and the products or services covered under their agreement would be the arms. News such as product announcements and reviews, bankruptcies, and layoffs as well as financial transactions like sales and purchases could be aggregated. This information would give an idea of the overall success of a business and would be helpful for investment purposes. Law enforcement agencies may also benefit from these techniques, as aggregating relationships could be useful in identifying entities doing business with suspected front companies for example. Searching for items other than

weapons, such as illegal drugs, or any item of interest could give law enforcement agencies a more complete picture and possibly allow them to perform their jobs more effectively.

In summary, we developed a novel technique for mining weapon transfer data. We demonstrated that it can process data faster than manual methods by calculating the performance on a set of data from the Web. We also showed a correlation between features types and relevant content, an adaptable approach to currency detection, and developed a method to visualize relations between entities mentioned in a weapons context. We showed that our technique is worthy of larger scale testing and development.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. SENTENCE EXCLUSIONS

```
Exclusions = ["acft", "actg", "adm", "admin", "adv", "af", "ala",
"alt", "a.m", "amp", "amph", "approx", "apr", "ariz", "asat",
"ass", "assn", "asso",
"assoc", "asst", "astro", "aug", "aux", "ave", "az", "b.a", "bal",
"bio", "bk", "bldg", "blvd", "bros", "b.s",
"btm", "c", "ca", "cal", "calif",
"capt", "cdr", "c.f", "ch", "chem", "cine", "civ", "cl", "co",
"col", "colo", "com", "comp", "conf", "conn", "coord",
"corp", "cpt", "crcl", "ct", "ctr", "ctrs", "cyn", "dec", "det",
"devel", "dia", "dir", "disp", "div", "dr", "drs", "dtd",
"e.g", "elev", "eng", "engn", "engr", "ens", "env", "exec",
"etc", "exp", "expl", "ext", "fac", "feb", "fig", "figs", "fl", "fla",
"flt", "fr", "frag", "frags", "freq", "ft", "ftr", "fwd", "gen", "geo",
"ghz", "gnd", "gov", "gy", "hex", "hgh", "hgr", "hist", "horiz",
"hosp", "hs", "hsg", "htz", "hwy", "ia", "i.e", "inc",
"incorp", "ind", "init", "inst", "instr", "int", "jan", "j.d", "jr",
"lic", "lt", "ltd", "m", "m.a", "ma", "maj",
"mass", "mav", "m.d", "mdl", "mech", "mfd", "mfg", "mic", "mics",
"mid", "mil", "minn", "mnt", "mods", "mr", "mrs", "m.s", "ms", "msec",
"msgr", "mt", "mtg", "mtn", "mtns", "mtr", "mts", "nano", "nat",
"nautic", "nav", "neg", "negs", "nev", "nov", "nv", "obd", "oct",
"off", "okla", "ops", "or", "ord", "org", "orig", "osc", "p", "pa",
"para", "pg", "pgm", "photog", "pkg", "pp", "pref", "prelim", "prep",
"pres", "ph.d", "p.m", "p.s", "p.p",
"prof", "prox", "pt", "pwr", "pyro", "qtr", "qtrs",
"qual", "rcvr", "rd", "recip", "ref", "refrig", "reg", "rel",
"rep", "repro", "ret", "rev", "rkt", "rm", "rnd", "rng", "roc", "rpt",
"scp", "sec", "secs", "sen", "sep", "sept", "ser", "serv", "sgt",
"sim", "sp", "spg", "sq", "sqd", "sqdn", "sqn", "sr", "st", "sta",
"std","supt", "surv", "swp", "sys", "tac", "tel", "temp", "tgt",
"thur", "topo", "trans", "transv", "tst", "twp", "twr",
"unkn", "unk", "u.k", "u.s", "u.s.a", "uv", "v", "va", "var",
"viz", "vol", "vs", "vt", "yrs"]
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. WEAPON LIST

The weapons listed here are provided in the format used by Python. This list was modified from the original list extracted from WordNet to remove items that were not of interest (e.g., “paintball gun”) or would create numerous false positives (e.g., “SAM”), and added others, which either were not included or were more general and would provide more matches (e.g., “missle”).

Weapons = ['matchlock', 'six-gun', 'Bren gun', 'sulfur mustard', 'automatic gun', 'fission bomb', 'flack', 'six-shooter', 'megaton bomb', 'A-bomb', 'high explosive', 'carbine', 'culverin', 'M-1', 'intercontinental ballistic missile', 'lachrymator', 'nerve agent', 'sarin', 'fusil', 'muzzle loader', 'spring gun', 'Bren', 'VX gas', 'semiautomatic pistol', 'musket', 'blistering agent', 'flying bomb', 'antiballistic missile', 'surface-to-air missile', 'Stinger', 'minute gun', 'Bofors gun', 'forty-five', 'nerve gas', 'pistol', 'Luger', 'nitrochloromethane', 'dichloroethyl sulfide', 'organophosphate nerve agent', 'chlorobenzylidenemalononitrile', 'gas gun', 'Browning machine gun', 'botulinus toxin', 'clean bomb', 'airgun', 'Colt', 'ballistic missile', 'MANPAD', 'Winchester', 'gas bomb', 'Bacillus anthracis', 'small-arm', 'antiaircraft gun', 'atomic bomb', 'automatic rifle', 'arquebus', 'Gatling gun', 'air-to-ground missile', 'pom-pom', 'doodlebug', 'biological weapon', 'CS gas', 'Minuteman', 'handgun', 'firelock', 'guided missile', 'side arm', 'heat-seeking missile', 'dirty bomb', 'chemical weapon', 'horse pistol', 'poison gas', 'space probe', 'mustard agent', 'Tommy gun', 'machine gun', 'submachine gun', 'autoloader', 'automatic firearm', 'M-1 rifle', 'Maxim gun', 'disrupting explosive', 'Spandau', 'soman', 'ack-ack gun', 'breechloader', 'air rifle', 'sniper rifle', 'chemical bomb', 'pepper spray', 'zip gun', 'automatic', 'clostridium perfringens', 'CN gas', 'flintlock', 'automatic weapon', 'Dragunov', 'anthrax bacillus', 'harquebus', 'self-loader', 'assault rifle', 'V-1', 'sidewinder', 'twenty-two', 'thermonuclear bomb', 'set gun', 'Sten gun', 'Quaker gun', 'air gun', 'nuclear weapon', 'whaling gun', 'firearm', 'aflatoxin', 'antiaircraft', 'horse-pistol', 'bioweapon', 'Mace', 'Very pistol', 'Garand rifle', 'SEB', 'tear gas', 'blunderbuss', 'repeater', 'ack-ack', 'machine pistol', 'revolver', 'derringer', 'rifle', 'atom bomb', 'air-to-air missile', 'shooting iron', 'flak', 'smoothbore', 'Mauser', 'botulismotoxin',

'Garand', 'assault gun', 'tabun', 'buzz bomb', 'neutron bomb', 'bursting explosive', 'lacrimator', 'burp gun', 'botulin', 'precision rifle', 'H-bomb', 'plutonium bomb', 'scattergun', 'ICBM', '.22', 'machine rifle', 'Kalashnikov', 'chloroacetophenone', 'teargas', 'semiautomatic', 'Browning automatic rifle', 'hydrogen bomb', 'robot bomb', 'repeating firearm', 'sawed-off shotgun', 'riot gun', 'light machine gun', 'twenty-two pistol', 'Verey pistol', 'Exocet', 'air-to-surface missile', 'fowling piece', 'semiautomatic firearm', 'automatic pistol', 'fusion bomb', 'ABM', 'Saturday night special', 'twenty-two rifle', 'hackbut', 'mustard gas', 'Chemical Mace', 'Uzi', 'hagbut', 'shotgun', 'Thompson submachine gun', 'staphylococcal enterotoxin B', 'brilliant pebble', 'cannon', 'peacekeeper', 'Peacemaker', 'atomic weapon', 'bioarm', 'nuke', 'WMD', 'weapon of mass destruction', 'ammunition', 'missile', 'weapon']

LIST OF REFERENCES

[1] U.S. Department of State. (2010). International traffic in arms regulations. [Online]. Available: http://pmddtc.state.gov/regulations_laws/itar_official.html

[2] U.S. Department of State. (2009, January). *The Arms Export Control Act.* [Online]. Available: http://pmddtc.state.gov/regulations_laws/aeca.html

[3] R.F. Grimmett, "Conventional arms transfers to developing nations, 2000–2007," Congressional Research Service, Washington, D.C., Congressional Report [RL34723], 2008.

[4] Federation of American Scientists (FAS). (2011). *The Arms Sales Monitoring Project*. [Online]. Available: http://www.fas.org/programs/ssp/asmp/index.html

[5] Stockholm International Peace Research Institute (SIPRI), (2011). *The financial value of the global arms trade*. [Online]. Available: http://www.sipri.org/research/armaments/transfers/measuring/financial_values

[6] United Nations (UN). (n.d.). *UN Register of Conventional Arms*. [Online]. Available: http://www.un.org/disarmament/convarms/Register/

[7] A. Gulli and A. Signorini, "The indexable web is more than 11.5 billion pages," in Special interest tracks and posters of the 14th international conference on World Wide Web. *ACM*, DOI=10.1145/1062745.1062789, pp. 902–903, 2005.

[8] Google. (2011). *Sizing Up Search Engines*. [Online]. Available: http://www.google.com/help/indexsize.html

[9] W3C Technical Architecture Group. (2004, December). *Architecture of the World Wide Web, Volume One*. [Online]. Available: http://www.w3.org/TR/webarch/

[10] IETF Network Working Group. (1999, June). *Hypertext Transfer Protocol — HTTP/1.1 RFC2616*. [Online]. Available: http://datatracker.ietf.org/doc/rfc2616/

[11] A. Arasu, J. Cho, H.Garcia-Molina, A. Paepcke, and S. Raghavan, "Searching the web," *ACM Trans. Internet Technol.*, pp. 2–43, August 2001.

[12] Free Software Foundation, Inc. (2010, November). *GNU Wget*. [Online]. Available: http://www.gnu.org/software/wget

[13] D. Stenberg. (2011, May). *cURL*. [Online]. Available: http://curl.haxx.se

[14] J. Gregario and A.Fackler. (2011). *Httplib2 - A comprehensive HTTP client library in Python*. [Online]. Available: http://code.google.com/p/httplib2/

[15] M. K. Bergman. (2001, August). White paper: The deep web: Surfacing hidden value. *The Journal of Electronic Publishing* [Online]. *Vol. 7(1)*. Available: http://quod.lib.umich.edu/j/jep/3336451.0007.104?rgn=main;view=fulltext

[16] J. Cho and H.Garcia-Molina, "Estimating the frequency of change," *ACM Trans. Internet Technol*., vol. 3 pp. 256–290, August 2003.

[17] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, pp. 107–117, 1998.

[18] M. Koster. (2010, August). *A Standard for Robot Exclusion*. [Online]. Available: http://www.robotstxt.org/orig.html

[19] D. Jurafsky and J. H. Martin, "Information extraction," in *Speech and Language Processing, Second Edition*. New Jersey: Pearson Education, 2009, ch. 22, sec. 1, pp. 727–734.

[20] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *Proceedings of the 14th conference on Computational linguistics - Volume 2* (COLING '92), Vol. 2, Stroudsburg, 1992, pp. 539–545

[21] U. Irmak and R. Kraft, "A scalable machine-learning approach for semi-structured named entity recognition," in *Proceedings of the 19th international conference on World wide web (WWW '10)*, New York, 2010, pp. 461–470

[22] S. Zhao, "Named entity recognition in biomedical texts using an HMM model," in *Association for Computational Linguistics, Geneva*, 2004, pp. 84–87.

[23] H. L. Chieu and H. T. Ng, "Named entity recognition with a maximum entropy approach," in *Association for Computational Linguistics, Edmonton,* 2003, pp. 160–163.

[24] J. Mayfield, P. McNamee, and C. Piatko, "Named entity recognition using hundreds of thousands of features," in *Association for Computational Linguistics, Edmonton*, 2003, pp. 184–187.

[25] Princeton University. (2011, February). *Wordnet a lexical database for English*. [Online]. Available: http://wordnet.princeton.edu/

[26] S. Bird. *Natural Language Toolkit*. [Online]. Available: http://www.nltk.org/

[27] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. Sebastopol, CA, US: O'Reilly Media, Inc., 2009.

[28] I. Horrocks, "Semantic web: The story so far," in *Proceedings of the 2007 International Cross-Disciplinary Conference on Web Accessibility (W4A)*. New York: ACM, 2007, pp. 120–125.

[29] W3C. (2009, October) *OWL 2 Web Ontology Language*. [Online]. Available: http://www.w3.org/TR/owl2-overview/

[30] International Organization for Standardization (ISO). (2011) *ISO 3166 code lists*. [Online]. Available: http://www.iso.org/iso/iso_3166_code_lists

[31] Amazon Web Services LLC. (2011). *Amazon Elastic Compute Cloud (Amazon EC2)*. [Online]. Available: http://aws.amazon.com/ec2/

[32] Neil C. Rowe and Kari Laitinen, "Semiautomatic disabbreviation of technical text," *Information Processing and Management*, vol. 31, no. 6, pp. 851–857, 1995

[33] Python Software Foundation. (2010, March). *re- Regular expression operations*. [Online]. Available: http://www.python.org/doc/current/library/re.html

[34] Stockholm International Peace Research Institute (SIPRI). (2011). *SIPRI Arms Transfers Database.* [Online]. Available: http://www.sipri.org/databases/armstransfers

[35] GlobalSecurity.org. (2011, July) *Congo Special Weapons*. [Online]. Available: http://www.globalsecurity.org/wmd/world/congo/index.html

[36] P. Alpers, and M. Wilson. (2011, October). *Guns in the Czech Republic: Facts, Figures and Firearm Law*. [Online]. Available: http://www.gunpolicy.org/firearms/region/czech-republic

[37] Ministry of Foreign Affairs of the Republic of Latvia. *Export Control of Strategically Significant Goods*. [Online]. Available: http://www.mfa.gov.lv/en/security/Directions/ExportControl/

[38] Ryerson University. (2003, September). *Tanzania*. [Online]. Available: http://www.ryerson.ca/SAFER-Net/regions/Africa/Tan_SR03.html

[39] P. Ruch, R. Baud, and A. Geissbuhler, "Toward filling the gap between interactive and fully-automatic spelling correction using the linguistic context," in *2001 IEEE International Conference on Systems, Man, and Cybernetics,* 2001, pp. 199–204.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Frederick Krenson
   SPAWAR Systems Center Atlantic
   Charleston, South Carolina